

REMARKS

Claims 1, 14 and 27 have been amended. Claims 1-28 remain pending in the application. Reconsideration is respectfully requested in light of the following remarks.

Section 112, Second Paragraph, Rejection:

The Examiner rejected claims 1-7, 12 and 25-28 under 35 U.S.C. § 112, second paragraph as indefinite. Applicants respectfully traverse this rejection for at least the following reasons.

Regarding claim 1, the Examiner asserts that the phrase “computer executable code built in to said device” is unclear and “fails to qualify how code is ‘built in’ to a given device, and what constitutes executable code being ‘built in’ to an arbitrary device”. Applicants respectfully traverse this rejection and assert that claim 1, when read in light of Applicants’ specification is clear and easily understood by one skilled in the art. Claim 1 further recites wherein the pre-generated message interface is constructed prior to runtime. Thus, one skilled in the art would easily understand the subject matter recited in claim 1. However, to expedite prosecution, 1, 14 and 27 have been amended to recite that computer-executable code built in to said device during a code-build process for the device. A code-build process is a well-understood and fundamental concept in the art of computer science and software/code application development. Applicants refer the Examiner to any elementary computer science text for further information on a code-build process in software development.

For at least the reasons above, removal of the § 112, second paragraph, rejection of claim 1 is respectfully requested. Similar remarks also apply to claims 14 and 27.

Regarding claims 12, 25, 26 and 28, none of these claims include the phrase, “computer executable code built in to said device” and thus the rejection of these claims

is improper. The Examiner has not provided any basis for the § 112, second paragraph, rejection of claims 12, 25, 26 and 28.

Section 102(e) Rejection:

The Examiner rejected claims 1-4, 8-17 and 21-28 under 35 U.S.C. § 102(e) as being anticipated by Weschler (U.S. Patent 6,842,903). Applicants respectfully traverse this rejection for at least the reasons below.

Regarding claim 1, Weschler fails to disclose **receiving an address for a service within the distributed computing environment and linking the address to a pre-generated message interface for accessing the service**, contrary to the Examiner's contention. Weschler teaches a method for providing dynamic references between services in a computer system that allows one service to obtain a reference to another service without requiring specific knowledge of the other service. In Weschler's system, pluggable modules may be used to provide additional functionality for a service and a service connector interface encapsulates the logic necessary to locate an instance of a particular service.

The Examiner cites column 8, lines 5 – 10. However, none of the cited passages, even considered in view of the rest of Weschler, discloses receiving an address for a service and linking that address to a pre-generated message interface for accessing the service. At column 8, lines 5 – 10, Weschler describes pluggable interfaces and pluggable modules that provide supporting functionality and implement program behavior for core profiling engine 201. Weschler describes that modules are “plugged in by specifying an initialization parameter” that “comprises an address or fully qualified path pointing to a location at which the plug-in module is stored.” However, the address described by Weschler is the address from which the plug-in model is loaded when instantiated (see, Weschler, column 8, lines 11-20).

Additionally, Weschler fails to disclose linking the address to a pre-generated message interface for accessing the service. The Examiner cites column 4, lines 15 – 30 and column 6, lines 20- 30 of Weschler. The cited passage describe Weschler's use of pluggable interfaces supporting connections to plug-in service modules. Weschler teaches that client applications access core profile engine 203 via a profile services API 203. API 203 "enables client applications that have a corresponding interface to send messages to request profile services from core profile engine 201. Weschler also teaches that core profile engine 201 supports methods for "gaining a reference to plug-in services".

However, Weschler fails to describe linking an address for a service to a pre-generated message interface for accessing the service. Firstly, the Examiner equates the address at which a plug-in module is stored with the address for a service of applicant's claim. However, the address at which a plug-in module is stored is not linked to a pre-generated message interface for accessing the service. Instead, Weschler teaches that the plug-in module may be stored locally or may not be stored on the same machine as core profile engine 201. In Weschler's system the plug-in module is loaded from the address at which it is stored and instantiated to provide additional functionality to core profile engine 201. Nowhere does Weschler mention linking the address at which a plug-in module is stored, which the Examiner equates to an address for a service, to a pre-generated message interface for accessing the service.

Applicants respectfully remind the Examiner that anticipation requires the presence in a single prior art reference disclosure of each and every limitation of the claimed invention, arranged as in the claim. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The **identical** invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). As discussed above, Weschler fails to disclose receiving an address for a service within the distributed computing environment and linking the address to a pre-generated message interface for accessing the service. Therefore, Weschler cannot be said to anticipate claim 1.

For at least the reasons above, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claims 14 and 27.

Regarding claim 2, Weschler fails to disclose a message interface of the message endpoint **verifying that the messages sent to the service comply with a message schema for the service**. The Examiner cites column 6, lines 25-50 of Weschler. However, the cited passage states that Weschler's core profile engine 201 provides a limited set of functions includes, among others, "management utilities for defining schemas." Weschler does not describe verifying that messages sent to the service comply with a message schema for the service. Weschler does not state that a schema managed via the functions provided by the core profile engine 201 is usable to verify messages sent to the service. Instead, Weschler teaches that core profile engine 201 provides management utilities for *defining* schemas. The fact that Weschler's core profile engine 201 provides functions for defining schemas has nothing do with, and fails to disclose, verifying that messages sent to a service comply with a message schema for the service.

A single mention of "management utilities for defining schemas", as cited by the Examiner, does not in any way disclose the specific limitation of verifying that message sent to a service comply with a message schema for the service. Thus, the rejection of claim 2 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claims 9, 15, and 22.

Regarding claim 8, the Examiner has failed to provide a proper *prima facie* rejection. The Examiner rejects claims 8-17 and 21-28 "on the same basis as Claims 1-4". However, claim 8 recites subject matter quite different from the subject matter of claims 1 – 4. Thus, the Examiner has failed to address the specific limitations of claim 8.

Further in regard to claim 8, Weschler fails to disclose **receiving a schema defining messages for accessing the service and generating message endpoint code according to the schema**. As noted above, the Examiner does not provide a proper rejection of claim 8. The Examiner does not cite any portion of Weschler that discloses the limitations of claim 8. Weschler teaches a method for providing dynamic references between services in a computer system that allows one service to obtain a reference to another service without requiring specific knowledge of the other service. Weschler teaches a service connector interface that encapsulates the logic necessary to locate an instance of a particular service.

Nowhere does Weschler mention anything about receiving a schema that defines messages for accessing a service, generating message endpoint code according to the schema, linking the message endpoint code into executable operating code for the device and loading the message endpoint code and operating code onto the device. In the rejection of claim 2, the Examiner cites column 6, lines 25 – 30 of Weschler and contends to Weschler's system includes a message schema. However, the cited passage states that Weschler's core profile engine 201 provides a limited set of functions includes, among others, "management utilities for defining schemas." However, a single mention of utilities for defining schema does not disclose the specific limitations of claim 1. Weschler makes no mention of the schema *defining messages for access the service*. Weschler also fails to mention anything *generating message endpoint code according to the schema*.

Thus, for at least the reasons presented above, the rejection of claim 8 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claim 14 and 27.

Section 103(a) Rejection:

The Examiner rejected claims 1-4, 8-17 and 21-28 under 35 U.S.C. § 103(a) as being unpatentable over Roberts et al. (U.S. Patent 6,560,633) (hereinafter “Roberts”) in view of Chen, et al. (U.S. Publication 2002/0062334) (hereinafter “Chen”). Applicants respectfully traverse this rejection for at least the following reasons.

Regarding claim 1, Roberts in view of Chen fails to teach or suggest **linking said address to a pre-generated message interface for accessing said service, wherein said pre-generated message interface is implemented by computer-executable code built in to said device during a code-build process for the device, wherein the pre-generated message interface is constructed prior to runtime.** Roberts teaches that the interfaces relied upon by the Examiner are downloaded from a web services engine and used by a client application during runtime (see, Roberts, column 4, line 60 – column 5, line 7, column 5, lines 19-30, and column 7, lines 10-33). Specifically, Roberts teaches a runtime model that “draws on the use of a number of web services that construct special user interface pages as output data” and that “[t]he behavior of these pages is to generate subsequent web service requests to the web services engine, and to call for the execution [of] action defined in a web services application session” (Roberts, column 4, line 67 – column 5, line 5). For example, Roberts describes that a web services request, “generates a response that takes the form of a graphical user interface to be displayed in a browser”. Roberts WSA interfaces are clearly meant to be downloaded and constructed *at runtime* and thus teach away from the use of computer-executable code built in to a device and comprising a pre-generated message interface for accessing a service, where the pre-generated message interface is constructed *prior* to runtime.

The Examiner relies on Chen to teach the use of dynamic agents to provide dynamic behavior modification of agents. Chen’s agents are designed to carry application specific actions, which can be loaded and modified on the fly (e.g. at runtime). Chen’s dynamic behavior modification allows a dynamic agent to adjust its capability for accommodating environmental and requirement changes. Chen’s dynamic

agents may also play different roles across multiple applications. The Examiner cites paragraph [0063] and refers to Chen's built-in APIs. However, the cited passage of Chen, even if combined with Roberts, does not teach or suggest computer-executable code built in to the device comprises the pre-generated message interface. Instead, Chen teaches built-in APIs that allows an action to create "a receiver thread and [to] register[] its socket address" (Chen, paragraph [0063]). APIs for creating threads and registering a socket address do not, even if combined with Roberts, teach or suggest a computer-executable code built into a device during a code-build process for the device and that comprises a pre-generated message interface for accessing the service.

In fact, nowhere does Roberts or Chen, whether considered singly or in combination teach or suggest linking said address to a pre-generated message interface for accessing said service, wherein said pre-generated message interface is implemented by computer-executable code built in to said device during a code-build process for the device, wherein the pre-generated message interface is constructed prior to runtime. Thus, even if combined as suggested by the Examiner, Roberts and Chen fail to teach or suggest the limitations of claim 1. Instead, the combination of Roberts and Chen, as suggested by the Examiner would result in a system using the WSA interfaces of Roberts but that also includes built in APIs for creating receiver threads and registering socket addresses.

For at least the reasons above, the rejection of claim 1 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claims 14 and 27.

The Examiner has failed to provide a proper rejection under § 103 of claim 8. The Examiner merely lists claim 8 as rejected under 35 U.S.C. 103(a), but fails to provide an actual rejection. However, claim 8 recites subject matter different from that recited by claim 1, the only claim for which the Examiner provides a rejection under 35 U.S.C. 103(a). Therefore, the rejection of claim 8 under § 103 is improper.

Further in regard to claim 8, Roberts in view of Chen fails to teach or suggest generating message endpoint code according to a schema defining messages for accessing a service, linking the message endpoint code into executable operating code for the device and loading the message endpoint code and operating code onto the device. The Examiner fails to cite any portion of Roberts or Chen that teaches or suggests linking message endpoint code, generated *according to a schema* defining messages for accessing a service, *into executable operating code* for a device and loading the message endpoint code and operating code onto the device. Instead, the Examiner merely relies upon Roberts' teachings regarding the web services architecture (WSA) generally, without providing any particular citation or interpretation that discloses *linking* message endpoint code *into executable operating code* for a device and loading the message endpoint code and operating code onto the device. For example, the Examiner cites column 5, lines 47-55, where Roberts describes that when a requester wants to run a WSA, an HTTP request is generated to run a "model-based" web service that has the responsibility of maintaining runtime models for WSAs. However, the cited passage makes not mention of linking message endpoint code into executable operating code for a device. Nor does the cited passage describe loading the message endpoint code and the operating code onto the device. Similarly, the remainder of the Examiner's cited passage, as well as of Roberts, fails to mention anything regarding linking message endpoint code into executable operating code for a device and loading the message endpoint code and the operating code onto the device.

Additionally, Chen, whether considered singly or in combination with Roberts, also fails to teach or suggest generating message endpoint code according to a schema defining messages for accessing a service, linking the message endpoint code into executable operating code for the device and loading the message endpoint code and operating code onto the device. The Examiner has not relied upon Chen or cited any portion of Chen regarding this limitation of claim 8. Thus, the Examiner's combination of Roberts and Chen clearly fails to teach or suggest the limitations of claim 8.

Furthermore, Roberts specifically teaches downloading WSA interfaces from web service devices for use on client devices to access WSAs. Downloading of message interface is clearly quite different from linking message endpoint code, generated according to a schema defining messages for accessing a service, *into executable operating code* for a device and loading the message endpoint code and operating code onto the device. Thus, **Roberts teaches away** from Applicants' claim.

For at least the reasons above, the rejection of claim 8 is not supported by the cited art and removal thereof is respectfully requested. Similar remarks also apply to claims 21 and 28.

Claims Objected To But Otherwise Allowable:

Claims 5-7 and 18-20 were objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form. In light of the above remarks, Applicants assert that these claims are allowable in their present form.

CONCLUSION

Applicants submit the application is in condition for allowance, and prompt notice to that effect is respectfully requested.

If any extension of time (under 37 C.F.R. § 1.136) is necessary to prevent the above-referenced application from becoming abandoned, Applicants hereby petition for such an extension. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-66200/RCK.

Also enclosed herewith are the following items:

- ☐ Return Receipt Postcard
- ☐ Petition for Extension of Time
- ☐ Notice of Change of Address
- ☐ Other:

Respectfully submitted,

/Robert C. Kowert/

Robert C. Kowert, Reg. #39,255
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8850

Date: November 21, 2006